

AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A program execution control device comprising a processor and a storage unit,

wherein said program execution control device causes said processor for causing a processor to execute a program stored in said storage unit,

wherein the program is composed of one or more sets of bytecodes including a bytecode for invoking a bytecode set,

wherein said program execution control device further comprises:

a judging unit operable, for each execution of an invocation bytecode during execution of the program, to judge whether a bytecode set targeted for invocation is already compiled to native code specific to said[[the]] processor;

a first unit operable, when the target bytecode set is judged to be uncompiled, to instruct said[[the]] processor so that the target bytecode set is sequentially interpreted and executed, and to issue a request to compile the target bytecode set to native code;

a second unit operable, when the target bytecode set is judged to be compiled, to instruct said[[the]] processor to execute native code resulting from the compilation; and

a third unit operable to instruct said[[the]] processor to compile a bytecode set indicated by a compilation request issued by said first unit, in parallel with (i) the bytecode interpretation and execution by said first unit as well as with and (ii) the native code execution by said second unit₂[[.]]

wherein said program execution control device operates under control of a multitask operating system,

wherein the compilation by said third unit is executed as a separate task from the bytecode execution by said first unit and the native code execution by said second unit,

wherein the tasks of said first and second units are assigned a higher priority level than a priority level assigned to the task of said third unit, and

wherein said program execution control device further comprises:

a priority information acquiring unit operable to acquire information showing a priority level of each bytecode set;

a comparing unit operable to compare priority levels of (i) the bytecode set indicated by the compilation request and (ii) the instruction execution task; and

a priority altering unit operable to temporarily raise a priority level of the compilation task, when the comparison shows that the priority level of the bytecode set indicated by the compilation request is higher than the priority level of the instruction execution task.

2. (Cancelled)

3. (Currently Amended) The program execution control device according to Claim 1[[2]], further comprising:

 a switching unit operable to switch to task execution by said third unit when task execution by said first or second unit is placed in a standby state.

4. (Currently Amended) The program execution control device according to Claim 3, further comprising:

 a request management unit operable to register compilation request information in said[[a]] storage unit in response to a compilation request issued by said first unit, and to manage the registered compilation request information, each piece of compilation request information being used for compiling a bytecode set indicated by a corresponding compilation request,

 wherein said third unit instructs said[[the]] processor to compile each bytecode set indicated by compilation request information registered in said[[the]] storage unit, in parallel with (i) the bytecode interpretation and execution by said first unit as well as with and (ii) the native code execution by said second unit.

5. (Currently Amended) The program execution control device according to Claim 3[[4]],

 wherein said request management unit places pieces of compilation request information in a queue in an order in which corresponding compilation requests, corresponding to the pieces of compilation request information, are received, and

wherein said third unit instructs said[[the]] processor to compile bytecode sets in order starting from a bytecode set indicated by a first piece of queued compilation request information.

6. (Currently Amended) The program execution control device according to Claim 4, wherein said request management unit does not register compilation request information in duplicate[[],] if compilation request information for a bytecode set indicated by a compilation request is already registered in said storage~~the storing~~ unit.

7. (Currently Amended) The program execution control device according to Claim 4, further comprising:

~~— a priority information acquiring unit operable to acquire information showing a priority level of each bytecode set, wherein~~

wherein said request management unit includes:

 a specifying subunit operable, in response to a compilation request issued by said first unit, to specify, with reference to the acquired priority information, a priority level of a bytecode set indicated by the compilation request;

 a comparing subunit operable to compare the specified priority level with a priority level of each bytecode set indicated by the registered pieces of queued compilation request information in said[[the]] storage unit; and

 a determining subunit operable to determine a position for placing a new piece of compilation request information for the bytecode set indicated by the compilation request[[],] so that the registered pieces of compilation request information are queued in descending order of priority.

8. (Currently Amended) The program execution control device according to Claim 4, further comprising:

 a relational information acquiring unit operable to acquire relational information showing each bytecode set together with [[all]]one or more bytecode sets related to the bytecode set; and

 a detecting unit operable to detect, with reference to the relational information, [[any]] one or more bytecode sets[[set]] related to the bytecode set indicated by the compilation request,

 wherein said request management unit registers compilation request information for the one or more related bytecode sets[[set]] detected by said detecting unit.

9. (Currently Amended) The program execution control device according to claim 4, ~~further comprising:~~

~~a priority information acquiring unit operable to acquire information showing a priority level of each bytecode set; wherein~~

wherein, with reference to the acquired priority information, said third unit instructs said[[the]] processor to compile bytecode sets indicated by the pieces of compilation request information registered in said[[the]] storage unit[[,]] in descending order of priority.

10. (Currently Amended) The program execution control device according to Claim 4, further comprising:

a count recording unit operable to keep a count of compilation requests made to a respective bytecode set when compilation of the bytecode set is repeatedly requested, and records the request count to said[[the]] storage unit as part of the corresponding compilation request information for the bytecode set; and

an acquiring unit operable to acquire a threshold of request count,

wherein said third unit instructs said[[the]] processor to compile bytecode sets in an order in which the respective request counts of the bytecode sets exceed the threshold.

11. (Currently Amended) The program execution control device according to Claim 5, further comprising:

a count recording unit operable to keep a count of compilation requests made to a respective bytecode set when compilation of the bytecode set is repeatedly requested, and records the request count to said[[the]] storage unit as part of the corresponding compilation request information for the bytecode set; and

an order altering unit operable to compare the respective request counts and alter positions of the pieces of queued compilation request information in descending order of the respective request counts.

12. (Currently Amended) The program execution control device according to Claim 5,

wherein said request management unit manages a plurality of queues with different priority levels, and

wherein said third unit instructs said[[the]] processor to compile bytecode sets in an order starting from bytecode sets indicated by compilation request information placed in a highest priority queue.

13. (Currently Amended) The program execution control device according to Claim 12, further comprising:

a special request information acquiring unit operable to acquire, prior to execution of the program, information showing a plurality of bytecode sets requested to be compiled,

wherein said request management unit registers, in a batch, compilation request information for all bytecode sets shown by the special request information[[,]] in a highest priority queue.

14. (Currently Amended) The program execution control device according to Claim 1, further comprising:

a second judging unit operable, when said judging unit judges that the target bytecode set is uncompiled, to judge whether the target bytecode set is currently under compilation; and

a fourth unit operable, when the target bytecode set is judged to be currently under compilation, to wait until the compilation is done, and to subsequently instruct said[[the]] processor to execute native code resulting from the compilation.

15. (Currently Amended) The program execution control device according to Claim 1, further comprising:

a request management unit operable to register compilation request information in a storage unit in response to a compilation request issued by said first unit, and to manage the registered compilation request information, each piece of compilation request information being used for compiling a bytecode set indicated by a corresponding compilation request,

wherein said third unit instructs said[[the]] processor to compile each bytecode set indicated by compilation request information registered in said[[the]] storage unit, in parallel with (ii) the bytecode interpretation and execution by said first unit ~~as well as with~~ and (ii) the native code execution by said second unit.

16. (Cancelled)

17. (Cancelled)

18. (Currently Amended) A program execution control method for causing a processor to execute a program composed of one or more sets of bytecodes including a bytecode for invoking a bytecode set, comprising:

a judging step, for each execution of an invocation bytecode during execution of the program, of judging whether a bytecode set targeted for invocation is already compiled to native code specific to the processor;

a first step, when the target bytecode set is judged to be uncompiled, of instructing the processor so that the target bytecode set is sequentially interpreted and executed, and of issuing a request to compile the target bytecode set to native code;

a second step, when the target bytecode set is judged to be compiled, of instructing the processor to execute native code resulting from the compilation; and

a third step of instructing the processor to compile a bytecode set indicated by a compilation request issued in said first step, in parallel with the bytecode interpretation and execution in said first step as well as with the native code execution in said second step. [[.]]

wherein the processor operates under control of a multitask operating system,

wherein the compilation by said third step is executed as a separate task from the bytecode execution by said first step and the native code execution by said second step,

wherein the tasks of said first and second steps are assigned a higher priority level than a priority level assigned to the task of said third step, and

wherein said program execution control method further comprises:

a priority information acquiring step of acquiring information showing a priority level of each bytecode set;

a comparing step of comparing priority levels of (i) the bytecode set indicated by the compilation request and (ii) the instruction execution task; and

a priority altering step of temporarily raising a priority level of the compilation task, when the comparison shows that the priority level of the bytecode set indicated by the compilation request is higher than the priority level of the instruction execution task.

19. (Cancelled)

20. (Currently Amended) A non-transitory computer readable recording medium having stored thereon a control program for causing a processor to execute another program composed of one or more sets of bytecodes including a bytecode for invoking a bytecode set, wherein, when executed, the control program causes the processor to perform a method comprising: ~~said control program comprising:~~

a judging step, for each execution of an invocation bytecode during execution of the program, of judging whether a bytecode set targeted for invocation is already compiled to native code specific to the processor;

a first step, when the target bytecode set is judged to be uncompiled, of instructing the processor so that the target bytecode set is sequentially interpreted and executed, and of issuing a request to compile the target bytecode set to native code;

a second step, when the target bytecode set is judged to be compiled, of instructing the processor to execute native code resulting from the compilation; and

a third step of instructing the processor to compile a bytecode set indicated by a compilation request issued in said first step, in parallel with the bytecode interpretation and execution in said first step as well as with the native code execution in said second step. [[.]]

wherein the processor operates under control of a multitask operating system,

wherein the compilation by said third step is executed as a separate task from the bytecode execution by said first step and the native code execution by said second step,

wherein the tasks of said first and second steps are assigned a higher priority level than a priority level assigned to the task of said third step, and

wherein said program execution control method further comprises:

a priority information acquiring step of acquiring information showing a priority level of each bytecode set;

a comparing step of comparing priority levels of (i) the bytecode set indicated by the compilation request and (ii) the instruction execution task; and

a priority altering step of temporarily raising a priority level of the compilation task, when the comparison shows that the priority level of the bytecode set indicated by the compilation request is higher than the priority level of the instruction execution task.